REMARKS

The Examiner makes certain requests regarding some of the materials provided in the Information Disclosure Statement. A new PTO-1449 form listing those references that the Examiner required additional information is attached with this response in addition to new copies of those references listed on the PTO-1449. A copy of the Examiner's initialed PTO-1449 form is respectfully requested.

Applicants note with appreciation the Examiner's withdrawal of the earlier grounds of rejection. However, claims 1, 2, 7, 11-14, and 16 now stand rejected under 35 U.S.C. §102(e) as being anticipated by Yates, U.S. Patent 6,502,237. This rejection is respectfully traversed.

To establish that a claim is anticipated, the Examiner must point out where each and every limitation in the claim is found in a single prior art reference. *Scripps Clinic & Research Found. v. Genentec, Inc.*, 927 F.2d 1565 (Fed. Cir. 1991). Every limitation contained in the claims must be present in the reference, and if even one limitation is missing from the reference, then it does not anticipate the claim. *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565 (Fed. Cir. 1986). Yates fails to satisfy this exacting standard.

Yates describes a computer system for executing a binary image conversion to convert instructions for a non-native computer system to a native computer system. A run-time system 32 (shown in Figures 2 and 3) collects "profile data" to determine execution characteristics of the non-native instruction. The non-native instructions and profiles statistics are fed to a binary translator operated in a "background mode." The non-native image is executed in two environments with a first portion executed as an interpreted image and remaining portions as a translated image. Yates defines an "interpreter" at column 2, beginning at line 62 through column 3, line 20, and a "translator" at column 3, lines 21-57.

The Examiner contends that Yates's run-time system 32 is the counterpart of the hardware execution unit of the independent claims, and the background system 34 is the counterpart of the claimed software execution unit. The independent claim integer (iv) recites:

> program instructions received by said hardware based execution unit for which execution is not supported by said hardware based execution unit are forwarded to said software based execution unit.

The Examiner refers to Yates at column 11, lines 9-25 as allegedly teaching this feature. Applicants disagree. Instead, Yates discloses in this text that the background system 34 translates non-native image segments into native image segments after the interpreter of the run-time system, i.e., the hardware execution unit, has finished execution of the instructions. Alternatively, the background system 34 operates during a pause in CPU utilization by the run time system 32. Accordingly, Yates fails to disclose forwarding instructions to the software based execution unit for which "*execution is not supported by*" the hardware based execution unit because the hardware execution unit, associated by the Examiner with the run-time system 32, executes the instructions before forwarding them to the software based execution unit, associated by the Examiner with Yates's background system 34.

The Examiner also refers to column 13, line 50 to column 14, line 3. The cited text in columns 13 and 14 specifies that native code instructions are executed. Then a determination is made based upon characteristics of a "return" instruction at the end of the section of native code as to whether the native image code can continue to be executed. Otherwise, control is transferred to the interpreter 44 of the run-time system 32. Contrary to what is recited in claims 1 and 16, integer (iv), Yates does not disclose that after a program instruction has been forwarded to the software based execution unit (background system 34), control is returned to the hardware based execution unit (run time system 32) for *the next program instruction* to be

executed. Indeed, Yates teaches away from this feature by specifying in column 13, lines 61-62, that the native code portion that is executed by the software execution unit includes "one or more basic blocks or routines of instructions." Rather than returning control for a next program instruction, the background system 34 continues to execute a plurality of program instructions in a sequence before control is returned to the hardware execution unit (run time system 32).

In relation to integer (v) of claims 1 and 16, the Examiner asserts that column 78, line 63 to column 79, lines 15-20 of Yates discloses the feature where scheduling support logic is provided in the hardware based execution unit to generate a scheduling signal for triggering a scheduling operation to be performed **between program instructions**. However, this cited portion of Yates does not disclose this feature. Yates describes a scheduling operation involving rearranging instructions into a more optimum sequence *during translation* of non-native code to native code. Since translation must be performed prior to execution of program instructions, it is clear that Yates does not teach triggering a scheduling operation "between program instructions," as specified by claims 1 and 16. This is further evidenced by Yates' teaching in column 68, lines 63-67, which make reference to scheduling optimization of code that ***will*** comprise the translated program code.

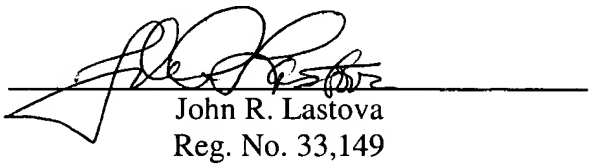
Thus, contrary to the Examiner's assertion, Yates does not disclose the features specified in integer (iv) or integer (v) of claims 1 and 16. The obviousness rejections that rely on APA in combination do not rectify Yates's basic deficiencies. The inventive technology defined by claims 1 and 16 enables a scheduling operation, like an interrupt, to be triggered irrespective of whether a preceding instruction has been executed by hardware or software. This scheduling is achieved by putting all of the program instructions through the hardware based execution unit, thus enabling the system to keep track of the instruction execution. Yates describes very

different scheduling operations related to rearranging sequences of instructions into a more optimal sequence during an instruction translation process from non-native program code to native program code. As explained above, the teachings of Yates would not motivate the ordinarily skilled person in this art to modify Yates's system to trigger scheduling operations *between* execution of program instruction.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 

John R. Lastova
Reg. No. 33,149

JRL:at
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100